Universidad Nacional de Ingeniería

Área de Conocimiento de Tecnología de la Información y Comunicación

Estimación de niveles de potencia para telefonía móvil en tecnología LTE basado en algoritmos de aprendizaje supervisado de

Trabajo Monográfico para optar al título de Ingeniero en Computación

1. INTRODUCCIÓN

Elaborado por:

Br. Maryin Abigail Palma Moraga
Carnet: 2011-39592

Tutor:

MSc. Ing. Cedrick Elknsherr Dalla Torre Parrales

Diciembre del 2023
Managua, Nicaragua

INDICE

Contenido	Página
1. INTRODUCCIÓN	1
2. JUSTIFICACIÓN	2
3. OBJETIVOS	3
a. Objetivo General:	3
b. Objetivos específicos:	3
4. MARCO TEORICO	4
4.1 Arquitectura General del Sistema LTE. [1]	5
4.2 Red de Acceso Evolucionado: E-UTRAN [1]	6
4.3 Red Troncal de Paquetes Evolucionada (EPC). [1]	7
4.4 IP Multimedia Subsystem (IMS).	10
4.5 Equipos de Usuario	11
4.6 Tecnologías de Nivel Físico. OFDMA, SC-FDMA y MIMO.	12
- **OFDMA.**	12
Ventajas de OFDMA	13
Desventajas de OFDMA	14
SC-FDMA	14
MIMO	15
5. MACHINE LEARNING	16
5.1 Distintos Tipos de Algoritmos de Machine Learning. [2]	16
5.2 Aplicaciones Prácticas de Machine Learning. [2]	17
5.3 Aprendizaje Supervisado	19
K-Nearest-Neighbor	20
El proyecto de ML. [5]	21

Introducción a PCA (Reducción de Dimensiones). [6] 25

6. CÓDIGO EN PYTHON 29

Tipos de datos de Pandas 30

La clase de objetos Series 31

REPRESENTACIÓN GRÁFICA DEL R-CUADRADO 36

7. CONCLUSIONES 39

8. RECOMENDACIONES 40

9. BIBLIOGRAFÍA 41

1. INTRODUCCIÓN

Es de suma importancia, para las empresas de telefonía móvil, tener datos que sirvan para hacer estimaciones. Estas estimaciones juegan un papel significativo, debido al número de clientes que tiende a subir por la gran variedad de servicios que ofrecen los operadores.

Los indicadores de calidad, describen en gran medida el comportamiento de cobertura en diferentes zonas del país. Estas zonas, se pueden clasificar en urbanas, suburbanas y rurales. A medida que han aparecidos nuevas tecnologías y aplicaciones, se debe satisfacer los niveles de potencia con que se radian diferentes entornos. Ya que el negocio de todas las empresas móvil, en gran medida, se basa en servicios multimedia y la velocidad tanto de carga como descarga que estos ofrecen.

Tener buenos niveles de cobertura para ofrecer los servicios que ofrecen los operadores, tiene un impacto en lo que respecta a la gestión de la red. Por ello, este trabajo monográfico, pretende hacer predicciones de niveles de potencia a partir de variables de entrada. De esa manera, se podrá estimar la pertinencia de una estructura para brindar el servicio de datos en una determinada área de interés.

Hoy en día, la industria 4.0, tiene como componente principal la inteligencia artificial, y la misma se basa en modelos de machine learning para hacer estimaciones en función de los requerimientos que se describen a partir de lo que se pretenda conocer, ya sea esto para darle solución a un problema, a una necesidad o bien a una oportunidad de hacer más eficiente una organización.

El aprendizaje supervisado tiene la intención de encontrar patrones en datos que se pueden aplicar a un proceso analítico que aporte significativamente en la toma de decisiones.

2. JUSTIFICACIÓN

Las redes móviles tienen a tener más clientes para el uso de servicios que los operadores ofrecen. En proceso de despliegue de infraestructura, para satisfacer la necesidad de cobertura en objetivos definidos por el área comercial, se debe conocer la pertinencia que podría tener una nueva estación base. Por ello, a partir de esta necesidad, en este proyecto monográfico se logra hacer estimaciones de los niveles de potencia, a partir de variables de entrada. Esto, sirve como insumo para determinar la ubicación más óptima para brindar el servicio de telefonía móvil en una determinada área, que podría ser, un entorno urbano, suburbano o rural.

3. OBJETIVOS

a. Objetivo General:

- Estimar niveles de potencia para telefonía móvil en tecnología LTE a partir de variables de entrada a partir de algoritmos de machine learning.

b. Objetivos específicos:

1. Realizar un estado del arte acerca de los trabajos de investigación que tienen relación con esta propuesta monográfica.

2. Hacer un estudio de los algoritmos de machine learning para hacer uso en lo que respecta a la estimación de los niveles de potencia en tecnología LTE.

3. Presentar un código en un lenguaje de programación (Python) para la realización de estimación de los niveles de potencia a partir de variables de entrada que se hayan identificado.

4. MARCO TEORICO

Arquitectura General de los Sistemas Celulares. [1]

Se pueden identificar tres elementos principales que constituyen la arquitectura de un sistema de comunicación celular:

- Equipo de usuario: Dispositivo que permite al usuario acceder a los servicios que nos ofrece la red. El dispositivo del usuario tendrá una tarjeta inteligente, que comúnmente denominamos tarjeta SIM (Subscribe Identity Module), que contendrá la información necesaria para poder conectarse a la red y poder disfrutar de los servicios que nos ofrece nuestro proveedor de servicio. Se conectará a la red a través de la interfaz radio.

- Red de acceso: es la parte del sistema que realiza la comunicación, transmisión radio, con los equipos de usuario para proporcionar la conectividad con la red troncal. Es la responsable de gestionar los recursos radio que estén disponibles para ofrecer los servicios portadores de una manera eficiente. La red de acceso está formada por estaciones base y dependiendo de la generación, por equipos controladores de estaciones base.

- Red troncal: parte del sistema que se encarga del control de acceso a la red celular, por ejemplo, la autenticación de los usuarios, gestión de la movilidad de los usuarios, gestión de la interconexión con otras redes, control y señalización asociada al servicio de telefonía, etc. Los equipos que conforman esta red albergan funciones de conmutación de circuitos, routing, bases de datos, etc.

4.1 Arquitectura General del Sistema LTE. [1]

En las especificaciones se denomina a la arquitectura del sistema LTE como Evolved Packet System (EPS). La idea es la misma que en las otras generaciones, dividir el sistema en los tres elementos mencionados anteriormente. Un equipo de usuario, una nueva red de acceso que denominaremos E-UTRAN y una red troncal que se denominará EPC. Todos los componentes que engloban este sistema están diseñados para soportar todo tipo de servicios de telecomunicación mediante mecanismos de conmutación de paquetes, por lo que no es necesario disponer de un dispositivo que trabaje en modo circuito, ya que en el sistema LTE los servicios con restricciones de tiempo real se soportan también mediante conmutación de paquetes. En la Figura 1. Se ve un ejemplo de la distribución de la arquitectura del sistema LTE.

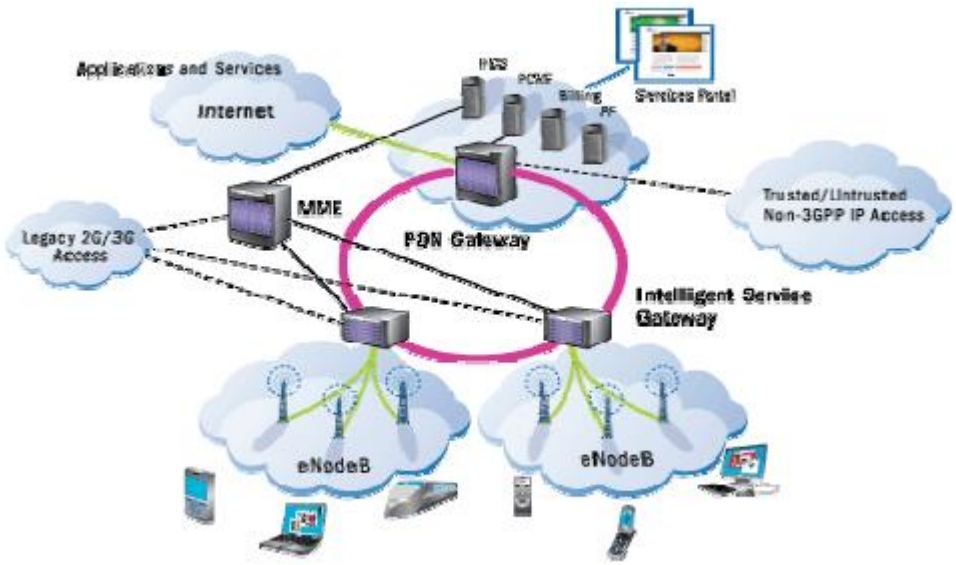


Figura 1. Esquema general de la arquitectura del sistema LTE. [1]

La red física que se utiliza en LTE para interconectar todos los equipos de la red, que se denomina red de transporte, es una red IP convencional. En la infraestructura de red LTE aparte de los equipos que realizan las funciones específicas del estándar, también habrá elementos de la red propios de redes IP como routers, servidores DHCP, servidores de DNS, switches, etc.

4.2 Red de Acceso Evolucionado: E-UTRAN [1]

En E-UTRAN la única entidad de red de en dicha red es la estación base, que en esta generación denominamos evolved NodeB (eNB). Esta estación base integra todas las funcionalidades de la red de acceso. Esto representa un cambio respecto a las anteriores generaciones, GSM y UMTS, ya que en éstas, la red de acceso contenía además de las estaciones base (BTS y NodoB), un equipo controlador (BSC y RNC). Esta diferencia se representa en la Figura 2.

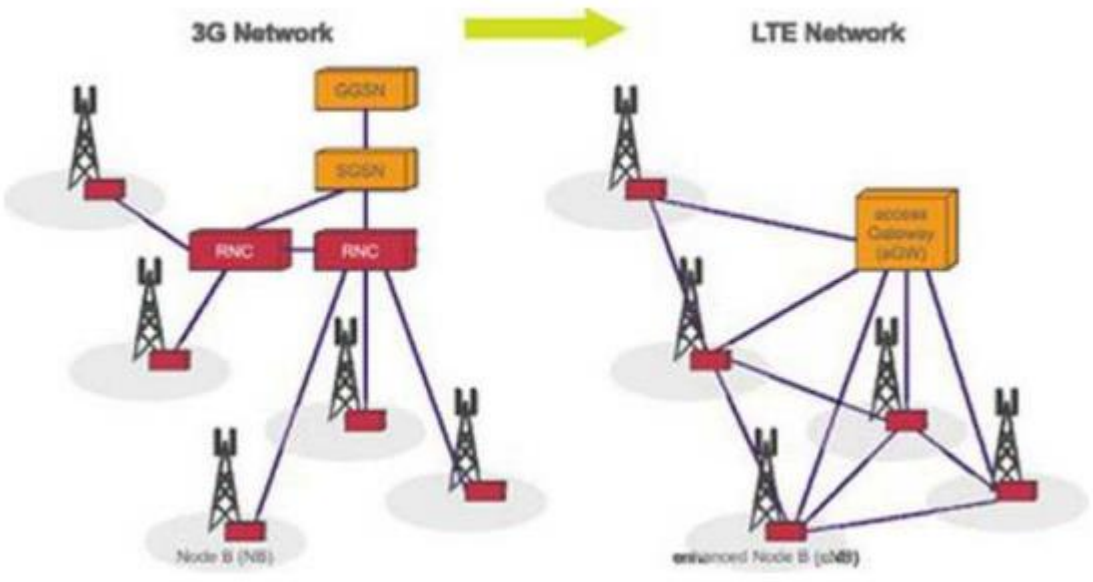


Figura 2. Comparación de la red de acceso entre 3G y 4G. [1]

El eNB tiene tres interfaces para comunicarse con los usuarios, con la red troncal y con otro eNB. E-UTRAN Uu es la interfaz radio que comunica al usuario con la estación base utilizando el canal radio.

Todas las funciones y protocolos que se necesitan para realizar el envío de datos y controlar la interfaz se implementa en la eNB. A la red troncal se comunica a través de la interfaz S1, que a su vez se divide en otras dos, la S1-MME, que se utiliza para el plano de control y S1-U para el plano de usuario. El plano de usuario se refiere a la torre de protocolos empleada para el envío de tráfico de usuario a través de dicha interfaz.

El plano de control se refiere a la torre de protocolos necesaria para sustentar las funciones y procedimientos necesarios para gestionar la interfaz. Esta separación entre las entidades de red, una dedicada al plano de usuario y otra al de control, nos permite dimensionar de forma independiente los recursos de transmisión necesarios para el soporte de la señalización del sistema y para el envío del tráfico de los usuarios.

La otra interfaz que existe es la X2, que se utiliza para conectar los eNBs entre sí. Gracias a esta interfaz se pueden intercambiar tanto mensajes de señalización, destinados a permitir una gestión más eficiente de los recursos radio, así como el tráfico de los usuarios del sistema cuando estos se desplazan de un eNB a otro en el momento de un traspaso (handover).

4.3 Red Troncal de Paquetes Evolucionada (EPC). [1]

Esta red ha sido concebida para proporcionar un servicio, como decíamos en la introducción, “all-IP”, es decir conectividad IP [2.4]. El núcleo de la red troncal EPC está formado por tres entidades de red, MME (Mobility Management Entity), Serving Gateway (S-GW) y el Packet Data Network Gateway (P-GW), que, junto a la base de datos principal del sistema denominada HSS (Home Subscriber Server), constituyen los elementos principales para la prestación del servicio de conectividad

IP entre los equipos de usuario conectados al sistema a través de la red de acceso E-UTRAN y redes externas a las que se conecta la red troncal EPC. Definimos a continuación cada una de estas entidades de red:

- MME: Es el elemento principal del plano de control de la red LTE para gestionar el acceso de los usuarios a través de E-UTRAN. Todo terminal que se encuentre registrado en la red LTE y sea accesible a través de E-UTRAN, tiene una entidad MME asignada. Esta elección de MME se realiza dependiendo de varios aspectos tales como la ubicación geográfica del terminal en la red, así como a criterios de balanceo de cargas. Las principales funciones de esta entidad son:
    - Autenticación y autorización del acceso de los usuarios, siempre a través de EUTRAN. o Gestión de los servicios portadores EPS (EPS Bearer Service). Esta entidad es la encargada de gestionar la señalización que se necesita para establecer, mantener, modificar y liberar los servicios portadores.
    - Gestión de movilidad de los usuarios en modo idle (son terminales que no tienen establecida ninguna conexión de control con E-UTRAN pero están registrados en la red LTE).
    - Señalización para el soporte de movilidad entre EPS y otras redes externas. - S-GW: es la pasarela del plano de usuario entre E-UTRAN y la red troncal EPC. Igual que en la entidad MME, todo usuario registrado en la red LTE tiene asignado una entidad S-GW en la red EPC a través de la cual transcurre su plano de usuario. Las características principales son: o Proporciona un punto de anclaje en la red EPC con respecto a la movilidad del terminal entre eNBs.
    - La funcionalidad de anclaje también se aplica a la gestión de la movilidad con las otras redes de acceso del 3GPP (UMTS y GSM). o Almacenamiento temporal de los paquetes IP de los usuarios en caso de que los terminales se encuentren en modo idle.

- Encaminamiento del tráfico de usuario. Esta entidad albergará la información y funciones de encaminamiento necesarias para dirigir el tráfico de subida hacia la pasarela P-GW que corresponda y el tráfico de bajada hacia el eNB. - PDN Gateway (P-GW): Es la encargada de proporcionar conectividad entre la red LTE y las redes externas. Por lo tanto, un paquete IP generado en la red LTE resulta “invisible” en la red externa, a través de la entidad P-GW, que hace de pasarela entre una red y otra. Un usuario tiene asignada como mínimo una pasarela P-GW desde su registro en la red LTE. Principales características de esta entidad de red: o Aplicación de reglas de uso de la red y control de tarificación a los servicios portadores que tenga establecidos el terminal.
- La asignación de la dirección IP de un terminal utilizada en una determinada red externa se realiza desde la pasarela P-GW que corresponda. o Actúa de punto de anclaje para la gestión de movilidad entre LTE y redes externas no 3GPP (WiMAX, WiFi, CDMA2000, etc.)
- El tráfico IP que transcurre por la pasarela P-GW es procesado a través de un conjunto de filtros que asocian cada paquete IP con el usuario y servicio portador EPS que corresponda.

- HSS: es la base de datos principal que almacena los datos de todos los usuarios de la red. La información almacenada es tanto lo relativo a la subscripción del usuario como lo necesario para la operatividad de la red. Esta base de datos es consultada y modificada desde las diferentes entidades de red encargadas de prestar los servicios de conectividad o servicios finales (desde el MME de red troncal EPC y también desde servidores de control del subsistema IMS, que explicaremos más adelante). La información almacenada en la HSS que podemos encontrar: identificadores universales del usuario, identificadores de servicio, información de seguridad y cifrado, información relacionada con la ubicación de un usuario en la red, etc. HSS se estandarizó en 3GPP R5 en base a la

integración de dos entidades definidas en redes GSM y que se denominan HLR y AuC, a las que se les han añadido funcionalidades adicionales necesarias para soportar el acceso y la operativa del sistema LTE.

4.4 IP Multimedia Subsystem (IMS).

Es un subsistema que proporciona los mecanismos de control necesarios para la prestación de servicios de comunicación multimedia que están basados en la utilización del protocolo IP a los usuarios de la red LTE.

La idea es desplegar una infraestructura constituida por una serie de elementos (servidores, base de datos, pasarelas) que se comunicarán entre sí mediante una serie de protocolos, la mayoría estándares del IETF, y que nos permiten ofrecer servicios de voz y video sobre IP, videoconferencia, mensajería instantánea, etc. El acceso a estos servicios por parte de los terminales de usuario se realiza a través de los servicios de conectividad que ofrece la red LTE. La prestación de estos servicios por parte del IMS pretende sustituir a medio-largo plazo los servicios equivalentes ofrecidos actualmente en modo circuito.

El modelo de prestación de servicio en base al subsistema IMS se estructura en tres capas: transporte, control y aplicación.

- Capa de transporte: representa la infraestructura de red IP, que depende de la tecnología de acceso, que nos proporciona el encaminamiento de los flujos IP entre terminales y demás elementos de la red.

- Capa de control: aquí se ubican los elementos especializados en la gestión de sesiones, como los servidores SIP, así como otros elementos específicos para la interacción con redes telefónicas convencionales (pasarelas VoIP, controladores, etc.).

- Capa de aplicación: en esta capa residen los servidores de aplicación que albergan la lógica y datos asociados a los diferentes servicios proporcionados a través de IMS. En esta capa también se presentan elementos ligados a otras plataformas de servicios como redes inteligentes.

- El establecimiento y liberación de sesiones a través del IMS se basa en el protocolo de señalización SIP complementándolo con una serie de extensiones adicionales. SIP es un protocolo que se concibió para el establecimiento y liberación de sesiones multimedia (telefonía, videoconferencia, etc.) sobre redes IP entre dos o más participantes. Gracias a la flexibilidad de SIP, ahora abarca una gama de aplicaciones mucho más extensa, mensajería instantánea, juegos distribuidos, control remoto de dispositivos, etc.

4.5 Equipos de Usuario

Es el equipo que permite al usuario conectarse a la red LTE y disfrutar de los servicios que nos proporciona a través de la interfaz radio. La arquitectura funcional de un equipo de usuario es la misma que se definió para GSM y UMTS.

El equipo de usuario (User Equipment, UE) contiene dos elementos básicos: un módulo de subscripción del usuario (SIM/USIM) y el terminal móvil propiamente dicho (Mobile Equipment, ME). A su vez, el SE ME considera dos entidades funcionales: la terminación móvil (MT) y el equipo terminal (TE). A continuación, definimos todos estos elementos.

- Módulo de subscripción de usuario: La SIM/USIM está asociada a un usuario y por tanto es quien le identifica dentro de la red independientemente del equipo móvil utilizado. La separación entre SIM y ME facilita que un usuario pueda cambiar de terminal sin necesidad de cambiar de identidad, de SIM.

- El equipo móvil (ME): en él se integran las funciones propias de comunicación con la red celular, así como las funciones adicionales que permiten la interacción del usuario con los servicios que ofrece la red.

4.6 Tecnologías de Nivel Físico. OFDMA, SC-FDMA y MIMO.

A continuación, se definen los fundamentos más importantes del nivel físico que se implementan en el sistema LTE y que permiten alcanzar mayores niveles de capacidad y eficiencia en el uso de los recursos radio que los sistemas predecesores. En el enlace descendente se usa la técnica de acceso múltiple denominada OFDMA (Orthogonal Frequency Division Multiple Access) y para el enlace ascendente, la técnica denominada CS-FDMA (Single Carrier Frequency Division Multiple Access). Al final del apartado describiremos también las estructuras de transmisión y recepción con múltiples antenas.

- **OFDMA.**

Aunque la modulación OFDM se analizará con mayor detenimiento en el capítulo siguiente, diremos que la técnica de acceso múltiple OFDMA que se utiliza en el enlace descendente en el sistema LTE ofrece la posibilidad de que los diferentes símbolos modulados sobre las subportadoras pertenezcan a usuarios distintos. Por tanto, es posible acomodar varias transmisiones simultáneas correspondientes a diferentes flujos de información al viajar en subportadoras diferentes.

Se consigue que un conjunto de usuarios, puedan compartir el espectro de un cierto canal para aplicaciones de baja velocidad. El acceso múltiple se consigue dividiendo el canal en un conjunto de subportadoras que se reparten en grupos en función de la necesidad de cada uno de los usuarios. El sistema se realimenta con las condiciones del canal, adaptando continuamente el número de subportadoras asignadas al usuario en función de la velocidad que éste necesita y de las condiciones del canal. Si la asignación se hace rápidamente, se consigue cancelar de forma eficiente las interferencias co-canal y los desvanecimientos rápidos.

Ventajas de OFDMA

- Diversidad multiusuario: La asignación de subportadoras se realiza de manera dinámica. Como el canal radio presentará desvanecimientos aleatorios en las diferentes subportadoras, y que serán independientes de cada usuario, se puede intentar seleccionar para cada subportadora el usuario que presente un mejor estado del canal, es decir, el que perciba una mejor relación señal a ruido. Con esto conseguiríamos una mayor velocidad de transmisión y una mayor eficiencia espectral. A esta manera de actuar se le denomina scheduling.

- Diversidad frecuencial: es posible asignar a un mismo usuario subportadoras no contiguas, separadas suficientemente como para que el estado del canal en las mismas sea independiente, lo que nos proporciona diversidad frecuencial en la transmisión de dicho usuario ante canales selectivos en frecuencia.

- Robustez frente al multitrayecto: en el capítulo 3 ahondaremos más en este tema, pero adelantar que gracias a la utilización del prefijo cíclico, esta técnica es muy robusta frente a la interferencia intersimbólica (ISI), resultante

de la propagación multitrayecto y se puede combatir la distorsión mediante técnicas de ecualización en el dominio de la frecuencia, que resultan menos complejas que las que se realizan en el dominio del tiempo.

- Flexibilidad en la banda asignada: Esta técnica de acceso múltiple nos proporciona una forma sencilla de acomodar diferentes velocidades de transmisión a los diferentes usuarios en función de las necesidades de servicio requeridas por cada usuario, simplemente asignando más o menos subportadoras a cada usuario.

- Elevada granularidad en los recursos asignables: Como estamos subdividiendo la banda total en un conjunto elevado de subportadoras de banda estrecha que se asignan dinámicamente a los usuarios, se dispone de una elevada granularidad a la hora de asignar más o menos recursos a cada uno, con lo que nos ayudará a acomodar servicios con diferentes requisitos de calidad.

Desventajas de OFDMA

- Elevada relación entre la potencia instantánea y la potencia media (PAPR).
- Susceptibilidad frente a errores en frecuencia.

SC-FDMA

En el sistema LTE se ha optado por utilizar la técnica OFDMA para el enlace descendente porque en la estación base se quieren técnicas que incrementan la complejidad computacional para reducir el PAPR de la señal OFDMA, y no es tan

crítica la eficiencia ni el coste de los amplificadores de potencia. Sin embargo, en el terminal del usuario sí que es crítico reducir el consumo de potencia y conseguir por lo tanto una gran eficiencia en el amplificador, por lo que se ha optado por una técnica de acceso de portadora única. SC-FDMA se basa en unos principios de transmisión muy similares a los de OFDM, pero en este caso se efectúa una precodificación de los símbolos que se van a transmitir previa al proceso de transmisión OFDM, lo que nos permitirá reducir las variaciones en la potencia instantánea,

MIMO

El sistema MIMO utiliza múltiples antenas tanto para recibir como para transmitir. Una transmisión de datos a tasa elevada se divide en múltiples tramas más reducidas. Cada una de ellas se modula y transmite a través de una antena diferente en un momento determinado, utilizando la misma frecuencia de canal que el resto de las antenas.

Debido a las reflexiones por multitrayecto, en recepción la señal a la salida de cada antena es una combinación lineal de múltiples tramas de datos transmitidas por cada una de las antenas en que se transmitió. Las tramas de datos se separan en el receptor usando algoritmos que se basan en estimaciones de todos los canales entre el transmisor y el receptor.

Además de permitir que se multiplique la tasa de transmisión (al tener más antenas), el rango de alcance se incrementa al aprovechar la ventaja de disponer de antenas con diversidad. La teoría de la capacidad inalámbrica, extiende el límite del teorema de Shannon, en el caso de la utilización de esta tecnología.

Este resultado teórico prueba que la capacidad de transmisión de datos y rango de alcance de los sistemas inalámbricos MIMO se puede incrementar sin usar más espectro de frecuencias. Este aumento es de carácter indefinido, simplemente utilizando más antenas en transmisión y recepción. MIMO requiere la existencia de un número de antenas idéntico a ambos lados de la transmisión, por lo que en caso de que no sea así, la mejora será proporcional al número de antenas del extremo que menos antenas tenga.

5. MACHINE LEARNING

El Machine Learning es una disciplina del campo de la Inteligencia Artificial que, a través de algoritmos, dota a los ordenadores de la capacidad de identificar patrones en datos masivos y elaborar predicciones (análisis predictivo). Este aprendizaje permite a los computadores realizar tareas específicas de forma autónoma, es decir, sin necesidad de ser programados. [2]

El término se utilizó por primera vez en 1959. Sin embargo, ha ganado relevancia en los últimos años debido al aumento de la capacidad de computación y al boom de los datos. Las técnicas de aprendizaje automático son, de hecho, una parte fundamental del Big Data. [2]

5.1 Distintos Tipos de Algoritmos de Machine Learning. [2]

Los algoritmos de Machine Learning se dividen en tres categorías, siendo las dos primeras las más comunes:

- Aprendizaje supervisado: estos algoritmos cuentan con un aprendizaje previo basado en un sistema de etiquetas asociadas a unos datos que les permiten tomar decisiones o hacer predicciones. Un ejemplo es un detector de spam que etiqueta un e-mail como spam o no dependiendo de los

patrones que ha aprendido del histórico de correos (remitente, relación texto/imágenes, palabras clave en el asunto, etc.). [2]

- Aprendizaje no supervisado: estos algoritmos no cuentan con un conocimiento previo. Se enfrentan al caos de datos con el objetivo de encontrar patrones que permitan organizarlos de alguna manera. Por ejemplo, en el campo del marketing se utilizan para extraer patrones de datos masivos provenientes de las redes sociales y crear campañas de publicidad altamente segmentadas. [2]
- Aprendizaje por refuerzo: su objetivo es que un algoritmo aprenda a partir de la propia experiencia. Esto es, que sea capaz de tomar la mejor decisión ante diferentes situaciones de acuerdo a un proceso de prueba y error en el que se recompensan las decisiones correctas. En la actualidad se está utilizando para posibilitar el reconocimiento facial, hacer diagnósticos médicos o clasificar secuencias de ADN. [2]

5.2 Aplicaciones Prácticas de Machine Learning. [2]

El Machine Learning es uno de los pilares sobre los que descansa la transformación digital. En la actualidad, ya se está utilizando para encontrar nuevas soluciones en diferentes campos, entre los que cabe destacar: [2]

Recomendaciones: permite hacer sugerencias personalizadas de compra en plataformas online o recomendar canciones. En su forma más básica analiza el historial de compras y reproducciones del usuario y lo compara con lo que han hecho otros usuarios con tendencias o gastos parecidos.[2]

Vehículos inteligentes: según el informe Automotive 2025: industry without borders de IBM, en 2025 ya veremos coches inteligentes en las carreteras. Gracias al aprendizaje automático, estos vehículos podrán ajustar la configuración interna (temperatura, música, inclinación del respaldo, etc.) de acuerdo a las

preferencias del conductor e, incluso, mover el volante solos para reaccionar al entorno. [2]

Redes sociales: Twitter, por ejemplo, se sirve de algoritmos de Machine Learning para reducir en gran medida el spam publicado en esta red social mientras que Facebook, a su vez, lo utiliza para detectar tanto noticias falsas como contenidos no permitidos en retransmisiones en directo que bloquea automáticamente. [2]

Procesamiento de Lenguaje Natural (PLN): a través de la comprensión del lenguaje humano, asistentes virtuales como Alexa o Siri pueden traducir instantáneamente de un idioma a otro, reconocer la voz del usuario e incluso analizar sus sentimientos. Por otro lado, el PLN también se utiliza para otras tareas complejas como traducir la jerga legal de los contratos a un lenguaje sencillo o ayudar a los abogados a ordenar grandes volúmenes de información relativos a un caso. [2]

Búsquedas: los motores de búsqueda se sirven del aprendizaje automático para optimizar sus resultados en función de su eficacia, midiendo la misma a través de los clics del usuario. [2]

Medicina: investigadores del Instituto de Tecnología de Massachusetts (MIT) ya utilizan el Machine Learning para detectar con mayor antelación el cáncer de mama, algo de vital importancia ya que su detección temprana aumenta las probabilidades de curación. Asimismo, también se utiliza con una alta eficacia para detectar neumonía y enfermedades de la retina que pueden provocar ceguera. [2]

Ciberseguridad: los nuevos antivirus y motores de detección de malware ya se sirven del aprendizaje automático para potenciar el escaneado, acelerar la detección y mejorar la habilidad de reconocer anomalías. [2]

Para el caso de la propuesta de investigación, se pretende hacer predicciones de cobertura por niveles de potencia, considerando variables como: distancia, entorno físico y altura de torre. Dicha predicción, se realizará a través de Aprendizaje Supervisado.

5.3 Aprendizaje Supervisado

Esta categoría de Machine Learning, se podría entender como algoritmos que "aprenden" de los datos introducidos por una persona. Por tanto, se necesita la intervención humana para etiquetar, clasificar e introducir los datos en el algoritmo. El algoritmo genera datos de salida esperados, ya que en la entrada han sido etiquetados y clasificados por alguien. Existen dos tipos de datos que pueden ser introducidos en el algoritmo: [3]

Clasificación: clasifican un objeto dentro de diversas clases. Por ejemplo, para determinar si un paciente está enfermo o si un correo electrónico es spam. [3]

Regresión: predicen un valor numérico. Sería el caso de los precios de una casa al escoger diferentes opciones o la demanda de ocupación de un hotel. [3]

Algunas aplicaciones prácticas de este tipo de Machine Learning:

- La predicción de coste de un siniestro en el caso de las compañías de seguros. [3]

- La detección de fraude bancario por parte de entidades financieras. [3]

- La previsión de avería en la maquinaria de una compañía. [3]

K-Nearest-Neighbor

Es un algoritmo basado en instancia de tipo supervisado de Machine Learning. Puede usarse para clasificar nuevas muestras (valores discretos) o para predecir (regresión, valores continuos). Al ser un método sencillo, es ideal para introducirse en el mundo del Aprendizaje Automático. Sirve esencialmente para clasificar valores buscando los puntos de datos “más similares” (por cercanía) aprendidos en la etapa de entrenamiento

Es un método que simplemente busca en las observaciones más cercanas a la que se está tratando de predecir y clasifica el punto de interés basado en la mayoría de datos que le rodean. Como se dijo anteriormente, el Lenguaje Supervisado, quiere decir que tenemos etiquetado nuestro conjunto de datos de entrenamiento, con la clase o resultado esperado dada “una fila” de datos.

Esto quiere decir que dicho algoritmo no aprende explícitamente un modelo (como por ejemplo en Regresión Logística o árboles de decisión). En cambio, memoriza las instancias de entrenamiento que son usadas como “base de conocimiento” para la fase de predicción.

Aunque, sencillo, se utiliza en la resolución de multitud de problemas, como en sistemas de recomendación, búsqueda semántica y detección de anomalías.

Como ventaja tiene sobre todo que es sencillo de aprender e implementar. Tiene como contras que *utiliza todo el dataset* para entrenar “cada punto” y por eso requiere de uso de mucha memoria y recursos de procesamiento (CPU). Por estas razones kNN tiende a funcionar mejor en datasets pequeños y sin una cantidad enorme de features (las columnas).

El cálculo de la distancia entre el item a clasificar y el resto de items del dataset de entrenamiento.

- Seleccionar los “k” elementos más cercanos (con menor distancia, según la función que se use).
- Realizar una “votación de mayoría” entre los k puntos: los de una clase/etiqueta que <<dominen>> decidirán su clasificación final.

Para decidir la clase de un punto es muy importante el valor de k, este terminará casi por definir a qué grupo pertenecerán los puntos, sobre todo en las “fronteras” entre grupos. Por ejemplo, se elegiría valores impares de k para desempatar (si las features que utilizamos son pares). No será lo mismo tomar para decidir 3 valores que 13. Esto no quiere decir que necesariamente tomar más puntos implique mejorar la precisión. Lo que es seguro es que cuantos más “puntos k”, más tardará nuestro algoritmo en procesar y darnos respuesta.

Las formas más populares de “medir la cercanía” entre puntos son la distancia Euclidiana (la “de siempre”) o la Cosine Similarity (mide el ángulo de los vectores, cuanto menores, serán similares). Recordemos que este algoritmo -y prácticamente todos en ML- funcionan mejor con varias características de las que tomemos datos (las columnas de nuestro dataset). Lo que entendemos como “distancia” en la vida real, quedará abstracto a muchas dimensiones que no podemos “visualizar” fácilmente (como por ejemplo en un mapa). [4]

El proyecto de ML. [5]

Primero definamos en grandes rasgos las diversas etapas que conforman el desarrollo de un proyecto de Machine Learning.

1. Análisis de Negocio
2. Infraestructura de IA
3. Ingeniería de Datos
4. Modelado
5. Implementación / Despliegue

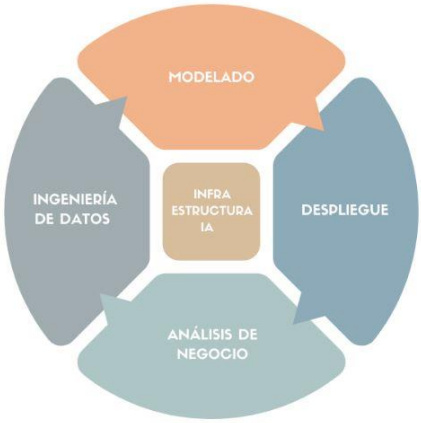


Figura 3. Ciclo de vida de un proyecto de machine learning. [5]

Siete Pasos de Machine Learning

Paso 1: Colectar Datos

Dada la problemática que deseas resolver, deberás investigar y obtener datos que utilizaras para *alimentar a tu máquina*. Importa mucho la calidad y cantidad de información que consigas ya que **impactará directamente en lo bien o mal** que luego funcione nuestro modelo. Puede que tengas la información en una base de datos ya existente o que la debas crear desde cero. Si es un pequeño proyecto puedes crear una planilla de cálculos que luego se exportará fácilmente como archivo csv. También es frecuente utilizar la técnica de web scraping para recopilar información de manera automática de diversas fuentes (y/o servicios rest/ APIs).

Paso 2: Preparar los datos

Es importante mezclar “las cartas” que obtengas ya que el orden en que se procesen los datos dentro de tu máquina no debe de ser determinante. También es un buen momento para hacer visualizaciones de nuestros datos y revisar si hay correlaciones entre las distintas características (“features”, suelen ser las columnas de nuestra base datos o archivo) que obtuvimos.

Habrá que hacer Selección de Características, pues las que elijamos impactarán directamente en los tiempos de ejecución y en los resultados, también podremos hacer reducción de dimensiones aplicando PCA si fuera necesario.

Se debe tener un balance para cada resultado(clase), para que sea representativo, ya que, si no, el aprendizaje podrá ser tendencioso hacia un tipo de respuesta y cuando nuestro modelo intente generalizar el conocimiento fallará.

También deberemos separar los datos en en dos grupos: uno para entrenamiento y otro para evaluación del modelo. Podemos fraccionar aproximadamente en una proporción de 80/20 pero puede variar según el caso y el volumen de datos que tengamos.

En esta etapa también podemos preprocesar nuestros datos normalizando, eliminar duplicados y hacer corrección de errores.

Paso 3: Elegir el modelo

Existen diversos modelos que podemos elegir de acuerdo al objetivo que tengamos: utilizaremos algoritmos de clasificación, predicción, regresión lineal, clustering (ejemplo k-means ó k-nearest neighbor), Deep Learning (ej: red neuronal), bayesiano, etc y podrá haber variantes si lo que vamos a procesar son imágenes, sonido, texto, valores numéricos.

Paso 4: Entrenar nuestra máquina

Se utilizará el set de datos de entrenamiento para ejecutar nuestra máquina y deberemos de ver una mejora incremental (para la predicción). Recordar inicializar los “pesos” de nuestro modelo aleatoriamente, los pesos son los valores que multiplican o afectan a las relaciones entre las entradas y las salidas, se irán

ajustando automáticamente por el algoritmo seleccionado cuanto más se entrena. Revisar los resultados obtenidos y corregir volver a iterar.

Paso 5: Evaluación

Deberemos comprobar la máquina creada contra nuestro set de datos de Evaluación que *contiene entradas que el modelo desconoce* y verificar la precisión de nuestro modelo ya entrenado. Si la exactitud es menor o igual al 50% ese modelo no será útil ya que sería como lanzar una moneda al aire para tomar decisiones. Si alcanzamos un 90% o más podremos tener una buena confianza en los resultados que nos otorga el modelo.

Paso 6: Parameter Tuning (configuración de parámetros)

Si durante la evaluación no obtuvimos buenas predicciones y nuestra precisión no es la mínima deseada es posible que tengamos problemas de overfitting (ó underfitting) y deberemos retornar al paso de entrenamiento (4) haciendo antes *una nueva configuración de parámetros de nuestro modelo*. Podemos incrementar la cantidad de veces que iteramos nuestros datos de entrenamiento (EPOCHs).

Otro parámetro importante es el conocido como “Learning Rate” (taza de aprendizaje) que suele ser un valor que multiplica al gradiente para acercarlo poco a poco al mínimo global (o local) para minimizar el coste de la función. No es lo mismo incrementar nuestros valores en 0,1 unidades que de 0,001 esto puede afectar significativamente el tiempo de ejecución del modelo.

También se puede indicar el máximo error permitido de nuestro modelo. Podemos pasar de tardar unos minutos a horas (y días) en entrenar nuestra máquina. A estos parámetros muchas veces se les llama Hiperparámetros. Este “tuneo” sigue siendo más un arte que una ciencia y se irá mejorando a medida que experimentamos.

Suele haber muchos parámetros para ir ajustando y al combinarlos se pueden disparar todas nuestras opciones.

Cada algoritmo tiene sus propios parámetros a ajustar. Por nombrar alguno más, en las Redes Neuronales Artificiales deberemos definir en su arquitectura la cantidad de hidden layers que tendrá e ir probando con más o con menos y con cuantas neuronas cada capa. Este será un trabajo de gran esfuerzo y paciencia para dar con buenos resultados.

Paso 7: Predicción o Inferencia

Cuando se tiene listo el para la utilización del modelo de Aprendizaje Automático con nueva información, se podrá hacer la realización de predicción o inferir resultados. [5]

Introducción a PCA (Reducción de Dimensiones). [6]

Imaginemos que queremos predecir los precios de alquiler de vivienda del mercado. Al recopilar información de diversas fuentes tendremos en cuenta variables como tipo de vivienda, tamaño de vivienda, antigüedad, servicios, habitaciones, con/sin jardín, con/sin piscina, con/sin muebles pero también podemos tener en cuenta la distancia al centro, si hay colegio en las cercanías, o supermercados, si es un entorno ruidoso, si tiene autopistas en las cercanías, la “seguridad del barrio”, si se aceptan mascotas, tiene wifi, tiene garaje, trastero… y seguir y seguir sumando variables.

Es posible que *cuanta más (y mejor) información, obtengamos una predicción más acertada*. Pero también empezaremos a notar que la ejecución de nuestro algoritmo seleccionado (regresión lineal, redes neuronales, etc.) empezará a tomar más y más tiempo y recursos. Es posible que algunas de las variables sean menos importantes y no aporten demasiado valor a la predicción. También podríamos acercarnos peligrosamente a causar overfitting al modelo.

Al quitar variables estaríamos haciendo Reducción de Dimensiones. Al hacer Reducción de Dimensiones (las características) tendremos menos relaciones entre variables a considerar. Para reducir las dimensiones podemos hacer dos cosas:

- Eliminar por completo dimensiones
- Extracción de Características

Eliminar por completo algunas dimensiones no estaría mal, pero *deberemos tener certeza* en que estamos quitando dimensiones poco importantes. Por ejemplo, para nuestro ejemplo, podemos suponer que el precio de alquiler no cambiará mucho si el dueño acepta mascotas en la vivienda. Podría ser un acierto o podríamos estar perdiendo información importante.

En la Extracción de Características si tenemos 10 características crearemos otras 10 características nuevas independientes en donde cada una de esas "nuevas" características es una combinación de las 10 características "viejas". Al crear estas nuevas variables independientes lo haremos de una manera específica y las pondremos en un orden de "mejor a peor" sean para predecir a la variable dependiente.

¿Y la reducción de dimensiones?

Como tenemos las variables ordenadas de "mejor a peores predictoras" ya sabemos cuáles serán las más y menos valiosas. A diferencia de la eliminación directa de una característica "vieja", nuestras nuevas variables son combinaciones de todas las variables originales, aunque eliminemos algunas, estaremos manteniendo la información útil de todas las variables iniciales.

¿Qué es Principal Component Analysis?

Entonces Principal Component Analysis es una técnica de Extracción de Características donde combinamos las entradas de una manera específica y podemos eliminar algunas de las variables "menos importantes" manteniendo la

parte más importante todas las variables. Como valor añadido, luego de aplicar PCA conseguiremos que todas las nuevas variables sean independientes una de otra.

¿Cómo funciona PCA?

En resumen, lo que hace el algoritmo es:

- Estandarizar los datos de entrada (ó Normalización de las Variables).
- Obtener los autovectores y autovalores de la matriz de covarianza.
- Ordenar los autovalores de mayor a menor y elegir los “k”. autovectores que se correspondan con los autovectores “k” más. grandes (donde “k” es el número de dimensiones del nuevo subespacio de características).
- Construir la matriz de proyección W con los “k” autovectores seleccionados.
- Transformamos el dataset original “X estandarizado” vía W para obtener las nuevas características k-dimensionales.

Todo esto ya lo hace solito scikit-learn (u otros paquetes Python). Ahora que tenemos las nuevas dimensiones, deberemos seleccionar con cuales nos quedamos.

Selección de los Componentes Principales

Típicamente utilizamos PCA para reducir dimensiones del espacio de características original (aunque PCA tiene más aplicaciones). Hemos rankeado las nuevas dimensiones de “mejor a peor reteniendo información”. Pero ¿cuantas elegir para obtener buenas predicciones, sin perder información valiosa? Podemos seguir 3 métodos:

Método 1:

Elegimos arbitrariamente “las primeras n dimensiones” (las más importantes). Por ejemplo, si lo que queremos es poder graficar en 2 dimensiones, podríamos tomar las 2 características nuevas y usarlas como los ejes X e Y.

Método 2:

calcular la *“proporción de* variación explicada*“* de cada característica e ir tomando dimensiones hasta alcanzar un mínimo que nos propongamos, por ejemplo hasta alcanzar a explicar el 85% de la variabilidad total.

Método 3:

Crear una gráfica especial llamada scree plot -a partir del Método 2- y seleccionar cuántas dimensiones usaremos por el método “del codo” en donde identificamos visualmente el punto en donde se produce una caída significativa en la variación explicada relativa a la característica anterior.

¿Por qué funciona PCA?

Suponiendo nuestras características de entrada estandarizadas como la matriz Z y ZT su transpuesta, cuando creamos la matriz de covarianza ZTZ es una matriz que contiene estimados de cómo cada variable de Z se relaciona con cada otra variable de Z. Comprender como una variable es asociada con otra es importante.

Los autovectores representan dirección. Los autovalores representan magnitud. A mayores autovalores, se correlacionan direcciones más importantes.

Por último, asumimos que a más variabilidad en una dirección particular se correlaciona con explicar mejor el comportamiento de una variable dependiente. Mucha variabilidad usualmente indica “Información” mientras que poca variabilidad indica “Ruido”.

Con PCA obtenemos:

1. Una medida de cómo cada variable se asocia con las otras (matriz de covarianza).

2. La dirección en las que nuestros datos están dispersos (autovectores).

3. La relativa importancia de esas distintas direcciones (autovalores).

PCA combina nuestros predictores y nos permite deshacernos de los autovectores de menor importancia relativa.

6. CÓDIGO EN PYTHON

Se considera una base de datos en Excel, se tiene información de entrada para considerar la tendencia del nivel potencia a partir de la coordenada para cada análisis de ubicación que se requiera. En la siguiente figura se muestra una imagen del archivo .csv a considerar.

1	2	3	4	5	6	7	8	9	10	11
ID	LATITUDE	LONGITUDE	Height	DISTANCIA	FACTOR DE Z	ALTURA mts	DISTANCIA n	NIVEL DE CO	COBERTURA	COVERAGE
1	12.13374	-86.35014	45	0 m	1	45	5	Excelente Ni	3	65
2	11.91603	-86.55058	45	4375 m	875	45	4375	Bajo Nivel de	1	152
3	12.17225	-86.35808	24	160 m	32	24	160	Excelente Ni	3	62
4	12.26923	-85.93385	42 m	325 m	65	42	325	Excelente Ni	3	63
5	11.88696	-86.43467	45 m	2843 m	568.6	45	2843	Bajo Nivel de	1	155
6	11.99014	-86.31067	42 m	34 m	6.8	42	34	Excelente Ni	3	60
7	12.08264	-86.26076	45 m	174 m	34.8	45	174	Excelente Ni	3	65

Figura 4. Datos a considerarse para hacer la predicción.

Se va a proceder a cargar las librerías a utilizar en la propuesta de código en Python para la estimación de niveles de potencia. A continuación, se muestran dichas librerías.

```
import sklearn as skl
import pandas as pd
from sklearn.linear_model import LinearRegression
import numpy as np

from sklearn.metrics import r2_score
```

Figura 5. Librerías a utilizar en el código.

Estas librerías permiten al programador tener acceso a varias funcionalidades de carácter específico, que ayudan significativamente a lograr las metas que tenga el desarrollador. Por ejemplo. Pandas es una librería que permite el manejo y análisis de estructuras de datos.

Las principales características de esta librería son: [7]

- Define nuevas estructuras de datos basadas en los arrays de la librería NumPy pero con nuevas funcionalidades.
- Permite leer y escribir fácilmente ficheros en formato CSV, Excel y bases de datos SQL.
- Permite acceder a los datos mediante índices o nombres para filas y columnas.
- Ofrece métodos para reordenar, dividir y combinar conjuntos de datos.
- Permite trabajar con series temporales.
- Realiza todas estas operaciones de manera muy eficiente.

Tipos de datos de Pandas

Pandas dispone de tres estructuras de datos diferentes:

- Series: Estructura de una dimensión.
- DataFrame: Estructura de dos dimensiones (tablas).
- Panel: Estructura de tres dimensiones (cubos).

Estas estructuras se construyen a partir de arrays de la librería NumPy, añadiendo nuevas funcionalidades.

La clase de objetos Series

Son estructuras similares a los arrays de una dimensión. Son homogéneas, es decir, sus elementos tienen que ser del mismo tipo, y su tamaño es inmutable, es decir, no se puede cambiar, aunque si su contenido.

Dispone de un índice que asocia un nombre a cada elemento de la serie, a través de la cual se accede al elemento. [7]

En este código, se está utilizando la herramienta Spyder, la que cuenta con una consola de entradas y salidas (i/o), a como se muestra en la Figura 6.

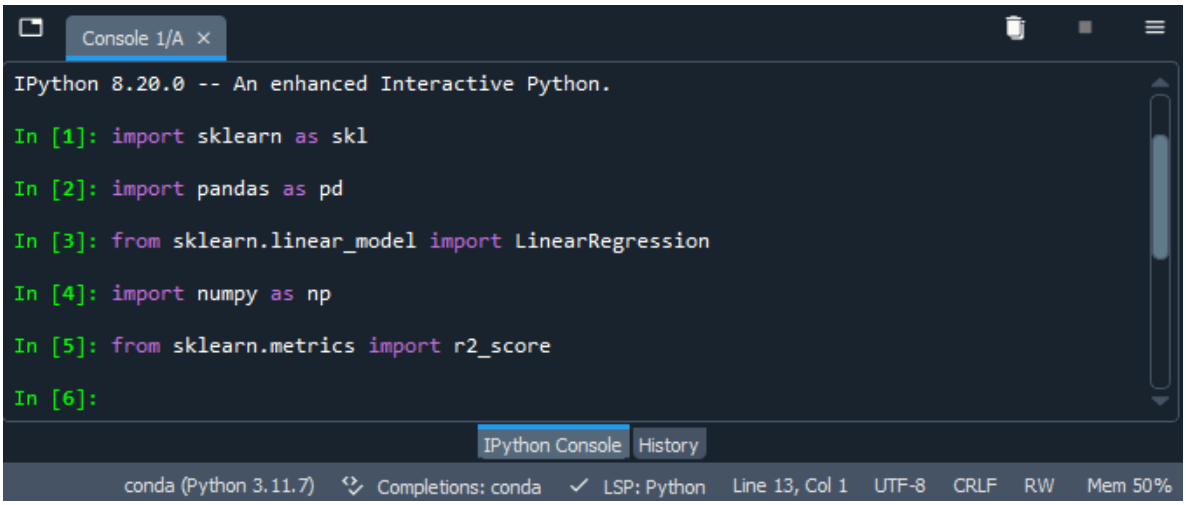


Figura 6. Consola de entrada y salida.

Se debe de proceder a cargar la base de datos, en este caso el nombre del archivo donde están los datos requeridos para lograr hacer la estimación de niveles de potencia. A continuación, en la siguiente figura, se muestran las líneas de programación que carga el archivo con los datos y se muestran los datos de las primeras 3 filas.

```
In [6]: data = pd.read_table('NIVEL.csv',sep=";")

In [7]: data.head(3)
Out[7]:
   ID  LATITUDE  LONGITUDE  ...              NIVEL DE COBERTA COBERTURA  COVERAGE
0   1  12.13374  -86.35014  ...  Excelente Nivel de Cobertura         3        65
1   2  11.91603  -86.55058  ...       Bajo Nivel de Cobertura         1       152
2   3  12.17225  -86.35808  ...  Excelente Nivel de Cobertura         3        62

[3 rows x 11 columns]
```

Figura 7. Primeros valores del archivo de los datos.

También, se puede mostrar los datos de la siguiente manera

```
In [9]: print(data)
      ID  LATITUDE  ...  COBERTURA COVERAGE
0      1  12.13374  ...          3       65
1      2  11.91603  ...          1      152
2      3  12.17225  ...          3       62
3      4  12.26923  ...          3       63
4      5  11.88696  ...          1      155
..   ...       ...  ...        ...      ...
119  121  13.09797  ...          2       82
120  122  12.05142  ...          1      110
121  123  12.01014  ...          3       60
122  124  13.64681  ...          3       60
123  125  13.69309  ...          1      160
```

IPython Console History

Figura 8. Impresión de los datos.

Además, es necesario la realización de una descripción estadística de los datos con que se cuentan, en la siguiente figura, se muestra el número de elementos, que en este caso son 124, la media, desviación estándar, el valor mínimo, valor máximo y cuartiles.

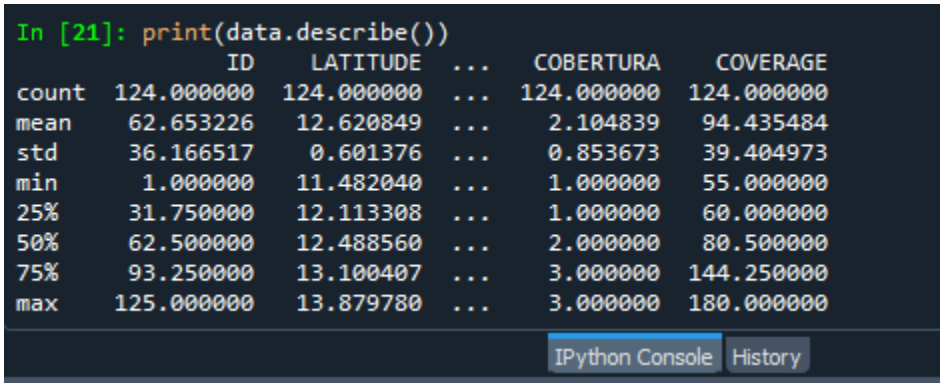


Figura 9. Descripción estadística de los datos.

A continuación, en la Figura 10, se muestran los datos únicos de la variable dependiente.

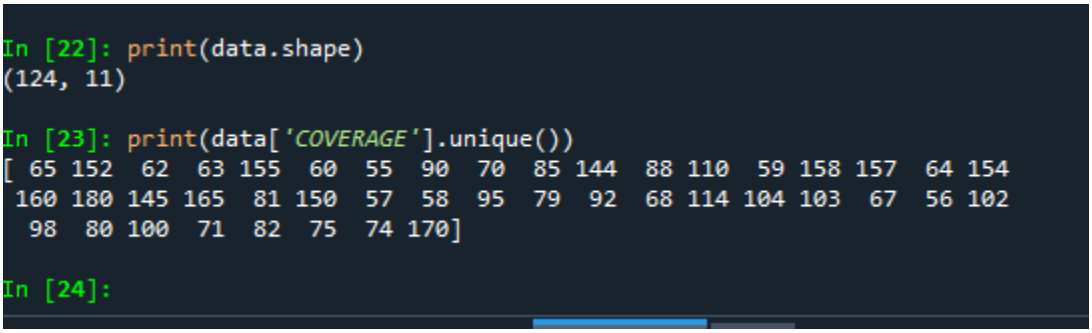


Figura 10. Datos únicos de la variable dependiente.

La frecuencia de los datos, para la variable dependiente, se muestra a través de la siguiente línea de programación. A como se muestra en la Figura 11.

Figura 11. Frecuencia de datos de la variable dependiente.

Para hacer la estimación, se debe de considerar tanto las variables dependientes y como dependiente de los datos obtenidos. A continuación, en la Figura 12, se muestra la asignación para cada una de las variables.

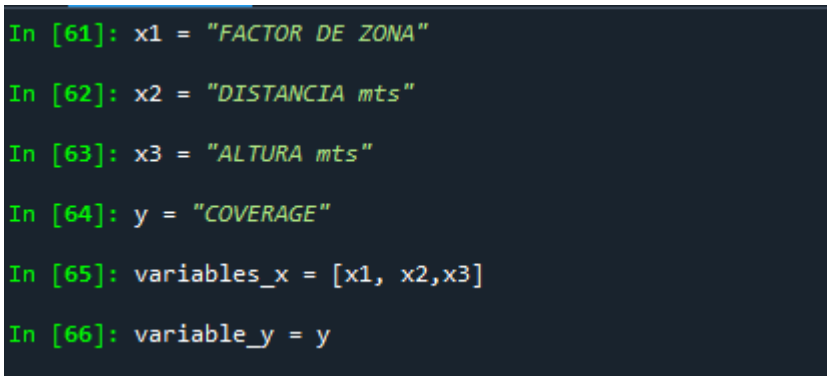


Figura 12. Asignación de variables.

Se tiene un modelo multivariado, ya que se están considerando 3 variables de entradas, estás variables dependen de la ubicación, una vez que se tiene la ubicación que se define mediante una coordenada geodésica, se requiere conocer la altura de la estructura, la distancia de separación entre el punto de medición y la estructura, y el tipo de zona. El factor de la zona está en dependencia del entorno, los entornos pueden ser: urbano, suburbano y rural.

Según un autor en [8], "*en regresión lineal simple se dice que se necesitan al menos 30 datos para que el teorema central del límite entre en vigor y las estimaciones sean consistentes, en regresión múltiple, se necesita además un número mínimo de casos en función de las variables a introducir. Se dice que, además de los 30 casos general se necesitan un mínimo de 10 casos por variable adicional (Si k es el número de variables independientes el mínimo sería de k+2 y algunos autores sugieren necesario k·20)*". En este caso, los casos y datos son lo mismo, como es un modelo de regresión lineal múltiple, se necesitan 30 datos + 10 por variable, pero se tienen 3 variables, entonces se necesitarían un total de 60 datos. En la base de datos, se disponen de 124, lo que indica que se sobre cumple.

En este caso, se tiene

A continuación, en la Figura 13, se muestra el tipo de modelo a utilizar, en este caso es una regresión lineal múltiple, se muestran los valores de coeficientes de las variables dependiente y el valor de intercepción.

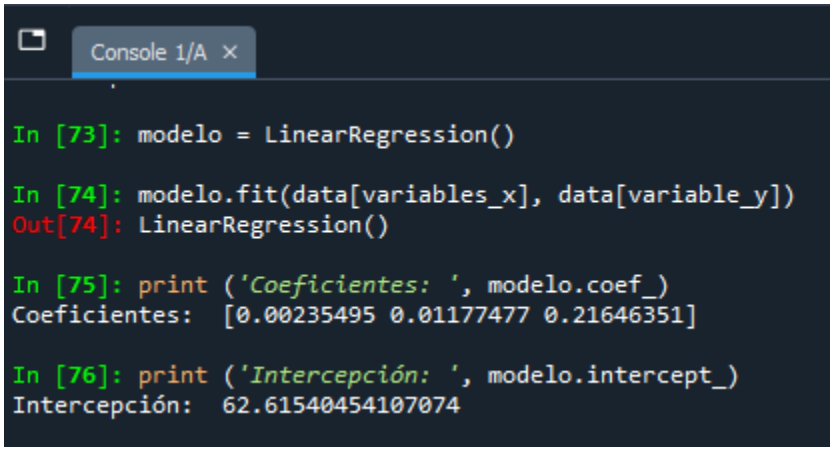


Figura 13. Modelo de regresión lineal múltiple.

También, es de suma necesidad generar la ecuación, para ello se deben considerar los valores de las variables del modelo. En la Figura 14, se muestra la ecuación.

```
In [78]: print('Ecuacion: y = {} * FACTOR DE ZONA + {} * DISTANCIA mts + {} * ALTURA mts +
{}'.format(round(modelo.coef_[0], 3), round(modelo.coef_[1], 3), round(modelo.coef_[2], 3),
round(modelo.intercept_, 3)))
Ecuacion: y = 0.002 * FACTOR DE ZONA + 0.012 * DISTANCIA mts + 0.216 * ALTURA mts + 62.615
```

Figura 14. Generando la ecuación.

El valor del coeficiente de determinación (R^2), nos brinda el grado de confianza de los datos que se estarán estimando, en este caso, el coeficiente determinación es 0.7357, a cómo se logra apreciar en la Figura 15, esto quiere decir que hay un 73.57% de confianza. Según [9], *tenemos "El R-cuadrado es una medida estadística de qué tan cerca están los datos de la línea de regresión ajustada. También se conoce como coeficiente de determinación, o coeficiente de determinación múltiple si se trata de regresión múltiple.*

La definición de R^2es bastante sencilla: es el porcentaje de la variación en la variable de respuesta que es explicado por un modelo lineal. Es decir:

$$R^2 = \frac{Variación\ explicada}{Variación\ total}$$

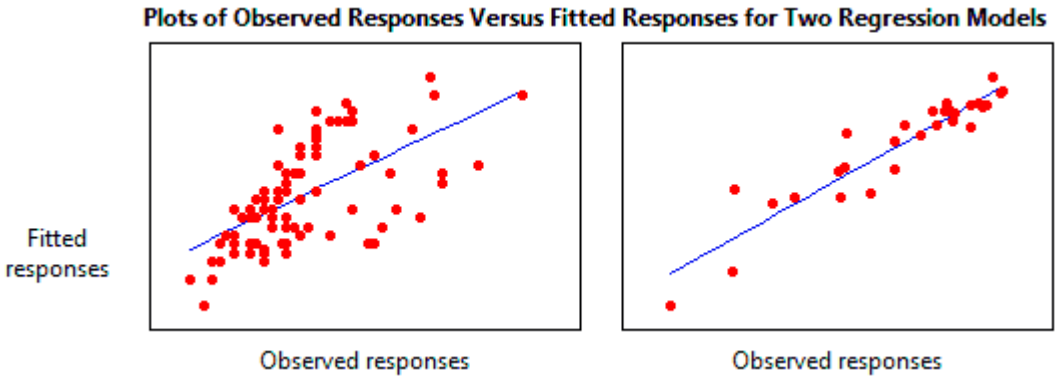
El R^2siempre está entre 0 y 100%:

- *0% indica que el modelo no explica ninguna porción de la variabilidad de los datos de respuesta en torno a su media.*
- *100% indica que el modelo explica toda la variabilidad de los datos de respuesta en torno a su media.*

En general, cuanto mayor es el R^2, mejor se ajusta el modelo a los datos. Sin embargo, hay condiciones importantes con respecto a esta pauta.

REPRESENTACIÓN GRÁFICA DEL R-CUADRADO

Representar gráficamente los valores ajustados en función de los valores observados ilustra diferentes valores del R^2para los modelos de regresión.



El modelo de regresión de la izquierda explica el 38% de la varianza, mientras que el de la derecha explica el 87,4%. Cuanto mayor sea la varianza explicada por el modelo de regresión, más cerca estarán los puntos de los datos de la línea de regresión ajustada. En teoría, si un modelo pudiera explicar el 100% de la varianza, los valores ajustados siempre serían iguales a los valores observados y, por lo tanto, todos los puntos de los datos estarían sobre la línea de regresión ajustada." [9]

```
In [25]: print('Coeficiente de determinación: ', round(r2_score(data[variable_y],
modelo.predict(data[variables_x])), 4))
Coeficiente de determinación:  0.7357
```

Figura 15. Modelo de determinación (R^2)

Para hacer una estimación, se requieren hacer la asignación de los valores de las variables de entrada, se pueden realizar de la siguiente manera. A como se muestra en la Figura 16.

```
Console 1/A ×

In [27]: FACTOR_NEW = 180

In [28]: DISTANCIA_NEW = 900

In [29]: ALTURA_NEW = 42

In [30]: prediccion_nueva = pd.DataFrame({x1: [FACTOR_NEW], x2: [DISTANCIA_NEW], x3:
[ALTURA_NEW]})
```

Figura 16. Asignación de los nuevos valores a las variables de entrada

En la Figura 17, se tendrá el resultado del nivel de potencia de cobertura, el cuál considera las 3 variables de entrada.

```
In [31]: COBERTURA_PREDICCIÓN = modelo.predict(prediccion_nueva)

In [32]: print('La PREDICCIÓN DE COBERTURA CON UN VALOR DE', FACTOR_NEW, ' para el factor de
Zona y un valor de distancia', DISTANCIA_NEW, 'Metros, tendrá un valor de cobertura en Dbm
de -', round(COBERTURA_PREDICCIÓN[0], 3))
La PREDICCIÓN DE COBERTURA CON UN VALOR DE 180  para el factor de Zona y un valor de
distancia 900 Metros, tendrá un valor de cobertura en Dbm de - 82.728
```

Figura 17. Valor estimado del nivel de potencia.

7. CONCLUSIONES

En este mundo digital, donde impera la necesidad de comunicarse, es de suma necesidad, la inversión por parte de los operadores de telefonía móvil en infraestructura que permita la eficiencia en lo que respecta al comportamiento de una red de telecomunicaciones inalámbrica. Entre mayor sea la precisión para la ubicación de estaciones bases, se podrá planear los objetivos de cobertura que se requieren para tener acceso a los servicios que ofrecen las empresas de telefonía celular.

La adquisición de datos, y su procesamiento, hace que se obtenga información con la cual puede servir perfectamente para la toma de decisiones, es clave que los datos sean confiables para que la estimación también lo sea. Y partiendo de eso haya un plan estratégico que sea de insumo para ir mejorando significativamente la calidad de cobertura de los operadores móviles.

Se logró, mediante modelos multivariados y la utilización de la programación un código en lenguaje de Python, hacer estimación de valores, a partir de tres variables de entrada. Dichas entradas son: El factor del entorno o zona, la distancia entre la infraestructura y del punto de interés, y la altura de la estructura, con esas variables se puedo hacer el cálculo del nivel de potencia de cobertura en una determinada localización.

8. RECOMENDACIONES

Se recomienda hacer uso de algoritmos de machine learning en el área de telecomunicaciones, ya que presenta un gran campo de aplicabilidad, y de necesidad para hacer propuestas que logren resolver problemas o necesidades.

Es importante incluir, otras técnicas de algoritmos de machine learning, a partir de las circunstancias que se presenten, para de esa manera para generar valor, tanto a nivel social, institucional y/o empresarial.

9. BIBLIOGRAFÍA

[1] LTE o la Cuarta Generación (4G) de Comunicaciones Móviles. Disponible en: https://biblus.us.es/bibing/proyectos/abreproy/11983/fichero/Cap%C3%ADtulo+2+-+LTE.pdf

[2] Qué es el Machine Learning. Disponible en: https://www.iberdrola.com/innovacion/machine-learning-aprendizaje-automatico

[3] Diferencias entre el Machine Learning Supervisado y no Supervisado. Disponible en: https://blog.bismart.com/diferencias-machine-learning-supervisado-no-supervisado

.[4] K-Nearest Neighbor. Disponible en:

https://www.aprendemachinelearning.com/clasificar-con-k-nearest-neighbor-ejemplo-en-python/

[5] Aprendizaje por refuerzo. Disponible en: https://www.aprendemachinelearning.com/

[6] Reducción de dimensiones. Disponible en: https://www.aprendemachinelearning.com/comprende-principal-component-analysis/

[7] La librería Pandas. Disponible en: https://aprendeconalf.es/docencia/python/manual/pandas/

[8] Montero Granados Roberto. “Modelos de Regresión Lineal Múltiples”. Departamento de Economía Aplicada. Universidad de Granada.

[9] Análisis de Regresión: ¿Cómo puedo interpretar el R-Cuadrado y Evaluar la Bondad de Ajuste? Disponible en: https://blog.minitab.com/es/analisis-de-regresion-como-puedo-interpretar-el-r-cuadrado-y-evaluar-la-bondad-de-ajuste